Using Causal Reasoning for Automated Failure Modes & Effects Analysis (FMEA)

Daniel Bell ◈ Martin Marietta Astronautics Group ◈ Denver
Lisa Cox ◈ Martin Marietta Astronautics Group ◈ Denver
Steve Jackson ◈ Martin Marietta Astronautics Group ◈ Denver
Phil Schaefer ◈ Martin Marietta Astronautics Group ◈ Denver

Key Words: Automated FMEA, Computer aided FMEA, FMECA, Artificial Intelligence, RAMCAD

SUMMARY & CONCLUSIONS

The growth of automated engineering analysis tools has been explosive. All of the major CAD/CAE platform manufacturers have assembled an array of tools to support the engineering analysis activity. These tools generally concentrate on the classical mathematical engineering analyses. We have developed a tool that automates the reasoning portion of a Failure Modes and Effects Analysis (FMEA). It is built around a flexible causal reasoning module that has been adapted to the FMEA procedure.

The approach and software architecture have been proven. A prototype tool has been created and successfully passed a test and evaluation program. We are expanding the operational capability and adapting the tool to various CAD/CAE platforms.

1. INTRODUCTION

This paper describes a tool that will reduce the cost of performing an FMEA by up to 90 percent, improve the accuracy of the analysis, and adapt to any analysis or report format and computing platform.

The requirement for an FMEA is common to both the private and public sectors. The methods and procedures for performing an FMEA are well defined. Historically, the analysis is performed by engineers hypothesizing failures of each component or subsystem or system in each of several failure modes and determining the effect on the operation of the component, subsystem, and system. It is a labor intensive activity. As such, the overall quality of the analysis can vary greatly, depending on the expertise and capability of the analyst.

Because the analysis procedure is repetitive, it lends itself to computer automation, but requires a program to perform the reasoning portion of the analysis. With the advent of initiatives such as Computer Aided Logistics Support (CALS) and Reliability and Maintainability Computer Aided Design (RAMCAD), several automated FMEA products have appeared on the market. Our research has shown that, in general, these tools do not automate the analysis portion of the FMEA, but are databases for storing and manipulating the FMEA data. An analyst must still manually create the data. The creation of the Multi-Purpose Causal tool (MPC) by the Martin Marietta Advanced Computing Technology group makes automation of the reasoning portion possible.

Our tool truly automates the analysis and is applicable to many different types of systems, from electronics to hydraulics.

We used a formal, structured method to create the tool. The steps were: (1) identify the requirements, (2) evaluate and select the best technical approach, (3) create a prototype program, in conjunction with expert analyst advisors, and (4) validate the approach through test.

2. ANALYZING THE REQUIREMENTS

Our project started with an in-depth evaluation of the requirements. What is an FMEA and what is it supposed to contain? This activity revolved around a detailed investigation and study of several governmental and industrial standards. A summary of the results is shown in Table 1. This task involved several disciplines, from safety, to design, to reliability and maintainability, to logistics engineering.

Table 1 lists the tasks and requirements associated with performing an FMEA and some additional tasks commonly performed that have synergism with the FMEA task. The table shows the FMEA effort broken into two basic groups, general requirements and detailed requirements. Each is discussed in the following paragraphs.

General Requirements

Under the general requirements there are three subgroups: implementation requirements, contributing or input information requirements, and documentation or output requirements. The implementation requirements provide general information and ground rules such as: how the FMEA will be performed, to what level it will be performed, and on what hardware it will be performed. The contributing or input requirements define the information needed by the designers, systems engineers, reliability engineers, safety engineers, and

0149-144X/92/0000-0343$01.00 © 1992 IEEE

other organizational elements. The documentation or output requirements of the FMEA task can take on many different forms depending on the applicable standards. The documentation of the information produced by the FMEA must provide the needed critical information without unneeded details. The tendency to produce large cumbersome documents makes this aspect of the FMEA a very critical part of the general requirements.

Detailed Requirements

The detailed requirements consist of the analytical portion of the FMEA. The FMEA takes a systematic approach for determining and evaluating each system, subsystem, part and component historical failure modes. This aspect of the FMEA is called the Failure Mode (FM) portion of the analysis. Once the FMs have been defined for the system and its hardware, the potential effects, or impact, of those failure modes on each part, subsystem, and system are evaluated according to the system safety of the mission, system performance of the mission, or the system maintenance of the mission. This portion of the FMEA analysis is the Effects Analysis (EA) portion. The FMEA can be extended to determine the criticality of the effects of each failure mode according to criticality criteria which involve the probability of the Failure Mode and the criticality of the effect. This Criticality Analysis (CA) task is often called out as a continuation of the FMEA.

All the tasks shown in Table 1 make up the body of the FMEA requirements used in the development of the prototype automated FMEA.

3. POSSIBLE TECHNIQUES FOR AUTOMATED FMEA

There are three primary technical approaches to performing automated FMEAs: numerical simulation, expert systems, and causal reasoning, each of which has advantages and disadvantages. An overview, with summary tradeoffs, is shown in Table 2.

Using Numerical Simulators

For most engineers, the first approach to automated FMEA that comes to mind is the use of numerical simulators. FMEAs could, in principal, be performed by replacing the normal equations for a component with a set of "failed" equations for that component. When the simulation was run with the injected fault, the behavior of the system could be predicted and reported in FMEA form. Clearly, a major advantage of this approach is that existing simulator technology, available in commercial-off-the-shelf (COTS) form, could be used.

However appealing and straightforward it may seem, upon closer inspection, some serious drawbacks to the numerical simulator approach are apparent. One drawback is in the definition of simulation equations to describe failed components. For example, consider the case of a resistor whose failure mode is "value drifted high." The implications of this failure could possibly vary widely depending on the actual amount of the value drift. If the resistor were part of a transistor biasing circuit, the effects might be negligible for small drifts. As part of a precisi n digital-to-analog (DAC) circuit, the same small drift c uld have disastrous effects. Most troublesome of all would be a circuit in which high or low drifts in resistance were tolerable, but for some small range of resistance values, instability would result. Clearly, then, it would not be appropriate to simply choose one or a fixed set of drifted values to model the "value drifted high" equations of the resistor, because one could never be sure if there were some other value in between that produced critical failure effects. More complex techniques such as interval analysis (Ref. 1) or Monte Carlo simulations would be needed for these failure modes.

Another drawback of the numerical simulation approach is in interpreting the failure effects after a simulation run. Simulators produce what is essentially a graph of numbers representing voltages, currents, and other electrical parameters. However, as discussed in the section "Analyzing Requirements," to avoid unnecessary detail, FMEA reports often need to be in qualitative, symbolic terms. Therefore, after going through the simulation phase, postprocessing software would need to compare the resulting numbers with the "normal" numbers and interpret the differences in terms of the effects. This is a complex undertaking; for example, determining whether an output waveform is a function of a particular input waveform is not possible by simple numerical comparisons.

The final drawback to the numerical simulation approach is run-time speed. Simulators can be quite slow, even for analysis of the normal case, because of the iteration for numerical convergence that is required (Ref. 2). To make matters worse, for an automated FMEA, the simulation would need to be run again for every failure mode of every component. This problem would be especially troublesome when performing analyses of systems containing large building blocks, such as analog microcircuits or microprocessors. In such cases, representing and simulating specific numerical faults would be prohibitive.

Table 1: APPLICABLE MIL-STD REQUIREMENTS, ANALYSES and DOCUMENTATION

Legend
X-REQUIRED
O-NOT REQUIRED
OP-OPTIONAL

	General Requirements																									Detailed Req.				
	IMPLEMENTATION INFORMATION AND OPTIONS												CONTRIBUTING/INPUT INFORMATION								DOCUMENTATION OUTPUT					TASK				
					TYPE		INDENTURE LEVEL (CONT. SPECS)																							
	FMECA PLAN	WORKSHEET	GROUND RULE	CODING SYS	FUNCTION	HARDWARE	LSA	CAT I,II	CAT III, IIII	FUNCTION	PIECE	SYSTEM	ENV. PROFILE	TECH SPECS	INTERFACE	DEV. PLAN	DESIGN DATA	DESIGN DRAW	REL. DATA	TRADE-OFF	FMECA REP.	SUMMARY	WORKSHEET	CRIT. ITEMS	TRADE REP.	FMEA	CA	FMEA MAIN	DMEA	FMECA PLAN
MIL-STD-785B, Reliability Program for System...																														
Task 204 FMECA	X	X	X	X	OP	OP	OP	OP	OP	OP	OP	OP	X	X	X	X	X	X	X	X	X	X	X	X	OP	X	X	OP	NA	X
Task 208 Reliability Critical Items	OP	OP	X	OP	OP	OP	OP	OP	OP	OP	OP	OP	X	X	X	X	X	X	X	X	OP	OP	OP	X	OP	X	X	OP	OP	OP
MIL-STD-1543B, Reliability Program for Space System																														
Task 204 FMECA	X	X	X	X	OP	OP	OP	OP	OP	OP	OP	OP	X	X	X	X	X	X	X	X	X	X	X	X	OP	X	X	OP	NA	X
Task 208 Reliability Critical Items	OP	OP	X	OP	OP	OP	OP	OP	OP	OP	OP	OP	X	X	X	X	X	X	X	X	OP	OP	OP	X	OP	X	X	OP	OP	OP
MIL-STD-1629A, Procedure for performing																														
Failure Modes, Effects and Criticality Analysis																														
General Requirements	X	X	X	X	OP	OP	OP	OP	OP	OP	OP	OP	X	X	X	X	X	X	X	X	X	X	X	X	X	OP	OP	OP	OP	X
Task 101 FMEA	X	X	X	X	OP	OP	OP	OP	OP	OP	OP	OP	X	X	X	X	X	X	X	X	X	X	X	X	X	X	O	O	O	X
Task 102 Criticality Analysis	X	X	X	X	OP	OP	OP	OP	OP	OP	OP	OP	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	OP	O	X
Task 103 FMECA-Maintainability Info.	X	X	X	X	OP	OP	OP	OP	OP	OP	OP	OP	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	O	X
Task 104 Damage Mode and Effects Analysis	X	X	X	X	OP	OP	OP	OP	OP	OP	OP	OP	X	X	X	X	X	X	X	X	X	X	X	X	X	X	O	O	X	X
Task 105 FMECA Plan	X	X	X	X	OP	OP	OP	OP	OP	OP	OP	OP	X	X	X	X	X	X	X	X	X	X	X	X	X	O	O	O	O	O
MIL-STD-470B, Maintainability Program for System...																														
Task 204 FMEA-Maintainability Information	X	X	X	X	OP	OP	OP	OP	OP	OP	OP	OP	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	O	OP
Task 207 Preparation of Inputs to the Detailed MP/LSA	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	X	O	X	O	O	O	O	X	OP	OP	OP	X	O	OP
MIL-STD-1591, C3 system and Component Fault Diagnosis																														
5.1.1 Contractual Requirements	O	O	O	O	O	O	O	O	O	O	O	O	X	X	X	X	X	X	X	OP	O	O	O	O	O	O	O	O	O	O
5.1.2 Primary System Configuration	O	O	O	O	O	O	O	O	O	O	O	O	O	X	X	X	X	X	X	OP	O	O	O	O	O	O	O	O	O	O
5.1.3 Reliability and FMEA Data	X	X	X	X	OP	OP	OP	OP	OP	OP	OP	OP	X	X	X	X	X	X	X	OP	OP	OP	X	X	X	X	O	O	O	OP
MIL-STD-2072, Survivability Aircraft; ... Program																														
5.2.2 Flight and Mission Essential Function	O	O	X	X	X	OP	OP	OP	OP	OP	OP	OP	O	X	X	X	X	X	O	OP	O	O	O	X	O	OP	O	O	OP	O
5.2.3 FMECA	X	X	X	X	OP	OP	OP	OP	OP	OP	OP	OP	X	X	X	X	X	X	X	X	X	X	X	X	OP	X	X	OP	NA	X
5.2.4 DMEA	X	X	X	X	OP	OP	OP	OP	OP	OP	OP	OP	X	X	X	X	X	X	X	X	X	X	X	X	X	X	O	O	X	X
MIL-STD-882B, System Safety...																														
Task 203 Subsystem Hazard Analysis	O	O	O	O	OP	OP	OP	OP	OP	OP	OP	OP	X	X	X	X	X	X	X	X	O	O	O	X	OP	X	X	O	O	O
Task 204 System Hazard Analysis	O	O	O	O	OP	OP	OP	OP	OP	OP	OP	OP	X	X	X	X	X	X	X	X	O	O	O	X	OP	X	X	O	O	O
MIL-STD-1388, Logistics Support Analysis																														
Task 301 Functional Requirements Identification	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	X	O	X	O	O	O	O	X	OP	OP	OP	X	O	OP
Appropriate DIDs																														
DI-L-7179, LSA-055, Failure Mode Detection Summary	X	X	X	X	OP	OP	OP	OP	OP	OP	OP	OP	X	X	X	X	X	X	X	X	X	X	X	X	X	X	O	O	O	X
DI-L-7176, LSA-052, Criticality Analysis Summary	X	X	X	X	OP	OP	OP	OP	OP	OP	OP	OP	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	OP	O	X
Other Applicable Analyses																														
Fault Tree Analysis	O	O	X	X	OP	OP	OP	OP	OP	OP	OP	OP	X	X	X	X	X	X	X	X	O	O	O	O	O	X	OP	O	O	O
Worst-Case Analysis	O	O	X	X	OP	OP	OP	OP	OP	O?	OP	OP	X	X	X	X	X	X	X	X	O	O	O	O	O	OP	OP	O	O	O
Part Stress Analysis	O	O	X	X	OP	OP	OP	OP	OP	OP	OP	OP	X	X	X	X	X	X	X	X	O	O	O	O	O	OP	OP	O	O	O

TABLE 2. POSSIBLE APPROACHES TO AUTOMATED FMEA

Approach	Advantages	Disadvantages
Numerical Simulation	• Precise numerical results • Use existing simulators	• Hard to produce precise fault models • Speed is often slow • Results must be translated into Symbolic terms
Expert Systems	• Could exploit 'Expert' knowledge of shortcuts • Translation much easier	• Almost impossible to cover all cases • Every system must be 'parsed' into categories known by expert system
Causal Reasoning	• Doesn't need precise models • Qualitative. Simulations fast • No need to enumerate all cases • Translation much easier	• Qualitative Math sometimes ambiguous

Using Expert Systems

An approach to automated FMEA which appears to address many of the above issues is to use expert systems. Here we define an expert system as software that captures "expert" task knowledge in the form of condition/action pairs, whether it be rule-based, object-oriented, etc. An FMEA expert system of this genre would encode expert knowledge of how failure modes of components affected the system, depending on how the component was placed in the topology of the system. For example, an expert system rule could say something like

```
IF the resistor is connected as
   EMITTER-BIAS
  AND the resistor drifts high in value

THEN the effect is that the emitter
   voltage of the associated
   transistor is too high
  AND the collector voltage is too low.

OR

IF the resistor drifts very high in
   value

THEN the effect is that the transistor
   is cut-off
  AND the collector and emitter
   voltages do not respond to inputs.
```

The development effort for such an approach would include capturing a large set of such failure mode and effect information; and developing techniques for parsing circuit information to determine things such as whether the resistor is connected in an EMITTER-BIAS configuration, propagating the effects in the rule to the outputs of the subsystem. The expert system approach would be good for overcoming some of the problems of numerical simulation. As demonstrated before, contextual information about how the component is used in the design could be encoded. This would alleviate the problems associated with choosing equations for faulted components, as well as the computational difficulties associated with running multiple simulations. Additionally, the rules in the expert system would be written using the same terminology that was needed in the FMEA report. The disadvantage, of course, is that it would be extremely challenging to reach the stage of knowing that ALL failure modes and topology information had been captured, or that the matching algorithms for determining the role of a component in the system could not be misled with a novel design.

Using Causal Reasoning

After examining these tradeoffs, we see that it would be most desirable to find an approach having the advantages of both numerical simulation and expert systems, while avoiding the shortcomings of each. This ideal approach would, like the simulator, directly use component models to derive behavior to guarantee not being misled by unforeseen connectivity of components. Like the expert system, however, it would not derive this behavior merely in terms of numbers and curves, but directly in terms of the language of the FMEA report. One approach that appears to satisfy this ideal is causal reasoning using qualitative models.

In the causal reasoning approach, components are described in terms of the cause-and-effect behavior they exhibit between inputs, outputs, and internal state (Ref. 3). The cause-and-effect behavior can be modeled numerically, but most often is described in discrete or symbolic terms. For example, one cause-effect relationship for a transistor is

```
When the transistor is in the LINEAR
   mode,

A Base Current of IB causes

A Collector Current of
(BETA * IB).
```

The causal reasoning system will directly consider the symbolic terms such as IB and (BETA * IB), rather than plugging "typical" numbers into the equations. In this way, it can SYMBOLICALLY reason about the effects of failures. For example, consider the failure mode of "decreased BETA". The causal reasoning system would find that the collector current of the transistor would now be

```
((BETA too-small) * IB).
```

By looking at this expression, using simple mathematical rules, the causal reasoner could derive that the collector current is too small. Downstream from the transistor further effects would be seen. Suppose that an output voltage in the circuit was found to be

$$\text{Vout} = \frac{(\text{R1} + \text{R2})\ \text{IB}}{((\text{BETA too-small}) * \text{IB})}$$

In this case, the causal reasoning system would similarly find that Vout was too large.

The causal reasoning approach overcomes some of the problems found with the previous approaches by directly reasoning with quantities such as "too-small" and "too-large." Because it does not need to commit to specific numerical values, the problems of modeling failure modes are greatly diminished. Additionally, because numbers are not used, the qualitative simulation avoids the issues of numerical convergence. At the same time, the approach retains the advantages of using component equations rather than rules of thumb. Causal reasoning also exhibits the major advantages of the expert system approach, in that reasoning is performed in terms of FMEA language (e.g., "too-small," "too-large"), avoiding the need for extensive postprocessing.

The causal reasoning approach can also be extended for building blocks larger than circuit components. For example, consider the application of FMEAs to microprocessor circuits. Containing hundreds of thousands of components, modeling even one of the common VLSI chips at the component level would be prohibitive, qualitatively or otherwise. For such applications, then, qualitative rules at a higher level of abstraction can be used. For example, causal relationships can be described representing transfer of data from registers to input or output ports, transformation of data by ALU instructions, and reading and writing to buses. Faults can be represented at a higher level also, by referencing things like stuck bits, or broken data links, without a need to refer to specific transistors or resistors. Because of these advantages, we selected the causal reasoning approach for our FMEA prototype.

However, there is a disadvantage to the approach which should be mentioned here. This is due to an inherent property of using qualitative mathematics. In some cases, causal reasoning will be unable to determine the net qualitative effect of opposing causes, for example, when adding opposite-signed numbers. In early qualitative reasoning systems (Ref. 3), this would have led to severe problems. However, recent advances in the state of the art (some of which were the results of our project), have significantly ameliorated these shortcomings. In the next section, we will discuss the causal reasoning approach, including the ambiguity issues, in more detail.

4. THE FMEA PROTOTYPE

The Causal Reasoning Approach

Causal reasoning is a technology for reasoning about complex systems in terms of their structure and behavior, rather than in terms of experiential rules of thumb, such as used by the expert systems approach (Ref. 4). To do this reasoning, a causal model of the target system is used directly, in lieu of software or rules that describe how to use the target system. For example, a causal reasoning diagnosis system will use the model to infer which failures could cause the observed symptoms, rather than using an explicit mapping from symptoms to failures. Intuitively, the goal of the causal reasoning approach is to encode how an engineer uses a diagram of a system, rather than to encode a set of situations and responses that the engineer previously encountered. The advantage of this approach is that it is not necessary to consider and solve all of the situations that may be encountered ahead of time. Rather, the software is given a diagram of the system (the causal model) and the means to use it to understand any situation that may arise (Ref. 3).

The MPC Causal Reasoning Tool

The Multi Purpose Causal (MPC) Tool is a general-purpose causal reasoning tool developed at Martin Marietta. It implements the causal reasoning approach with the architecture shown in Figure 1. MPC consists of three major subsystems: the causal model; a graphics interface with which the user can interact with the model, and the target system; and the causal reasoning modules. The reasoning modules implemented to date include simulation, command generation, fault detection, fault isolation, and FMEA. These modules are written to be target-system independent. Thus, to use MPC for a new application, all that is needed is the construction of a causal model of the new system, the reasoning modules are completely reusable. Additionally, after a model has been constructed for use by one of the reasoning modules, it can be used by any of the others. This approach has been applied to a variety of systems ranging from a few to many thousands of components.

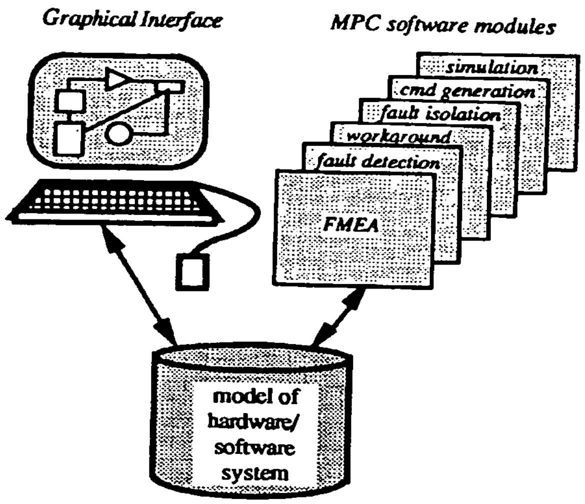
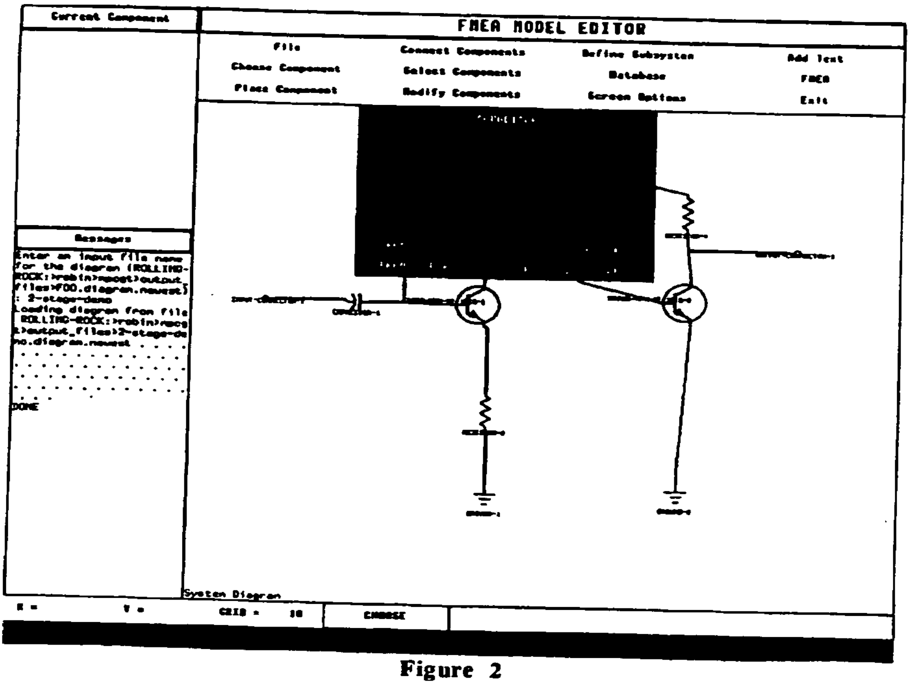


Figure 1

Building a Causal Model for FMEA

To perform an automated FMEA with MPC, a causal model of the target system must be constructed, then the FMEA reasoning module can perform the analysis. This section describes how model construction is accomplished.

The most natural way for an engineer to describe an engineered system is by using a diagram. Therefore, the MPC tool allows for models to be constructed directly from graphical diagrams of the system. System diagrams can be constructed at the piece-part, subsystem, or system level. At each level, different graphic icons are available to the engineer. For example, Figure 2 shows model construction taking place at the piece part level. Icons corresponding to various electronic components are available and can be connected to form a circuit.



Enter an input file name for the diagram (ROLLING-ROCK:>robin>mpcgt>output_files>FOO.diagram.newest) : 2-stage-demo
Loading diagram from file ROLLING-ROCK:>robin>mpcgt>output_files>2-stage-demo.diagram.newest
DONE

Figure 2

Each graphic icon in the diagram corresponds to a set of cause-effect mechanisms in the causal model. These mechanisms are represented internally as a set of symbolic equations. For example, consider the resistor shown in Figure 3a. The MPC equations for this resistor are:

```
When Resistor is NORMAL:
```

$$IT1 = \frac{T1 - T2}{RESISTANCE}$$

$$IT2 = IT1$$

More complex analog components may be described using qualitative differential equations (Ref. 5). Consider the capacitor shown in Figure 3b. The actual differential equation describing the capacitor is

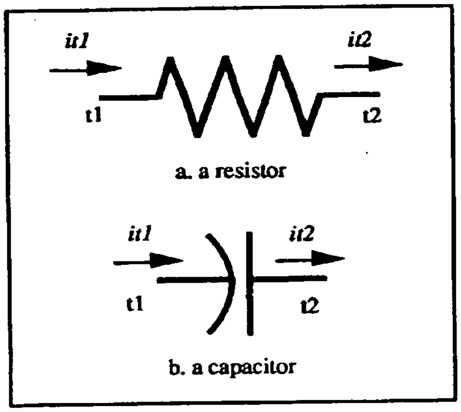
```
It1= It2 =
Capacitance d(t1 - t2)/dt
```

and the corresponding MPC equations are:

```
when Capacitor is NORMAL:

It1 = Capacitance *
        :deriv (t1 - t2)

It2 = It1
```



a. a resistor

b. a capacitor

Figure 3

Each component also corresponds to a set of equations describing failure modes. For example, one failure mode for a resistor is OPEN. When the resistor is OPEN, no current flows through it. The appropriate MPC equations are:

```
When Resistor is OPEN:

IT1 = 0
IT2 = 0
```

Similarly, one failure mode for the capacitor is SHORTED. This is expressed mathematically as the two terminals having the same voltage. The MPC equations are:

```
When Capacitor is SHORTED:

T1 = T2
```

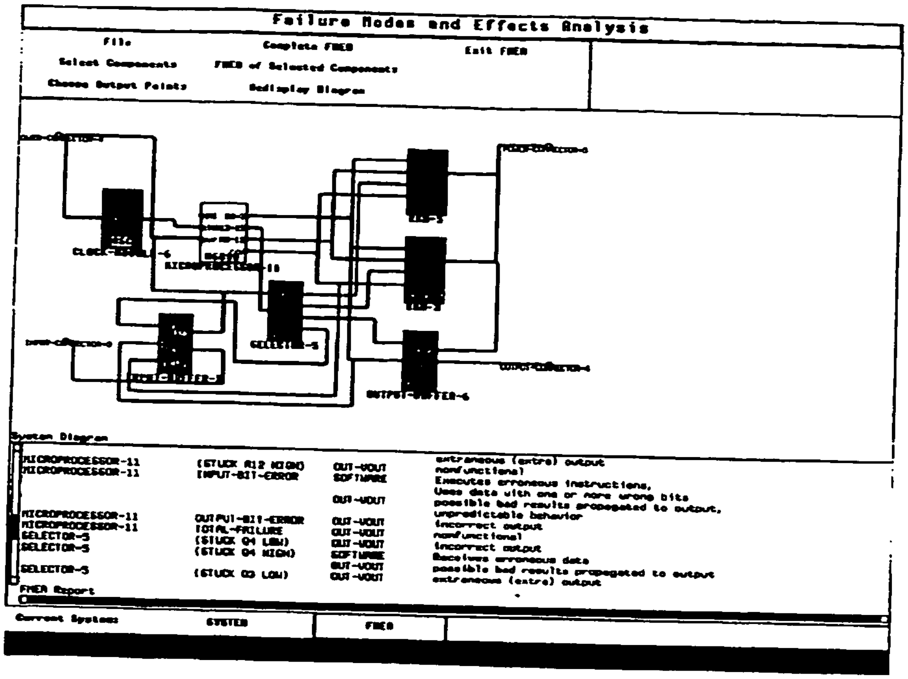
As the engineer places each component into the diagram, the corresponding equations are plugged into the evolving causal model. When the diagram is complete, MPC has a set of interrelated equations covering all parameters from the inputs of the system to the outputs. The final step in model generation is the assignment of cause and effect directions to the equations. The input terminals of the system are constrained to be causes of the corresponding equations, and the outputs are constrained to be effects. MPC uses an extension of the theory of causal ordering originally used in Econometrics (Ref. 6) to assign cause and effect roles to the parameters in the other equations, so that the entire system is described as a cause/effect flow from inputs to outputs.

Performing the FMEA

After a causal model has been constructed from the diagram, an automated FMEA is performed. The interface which the analyst sees is shown in Figure 4. In the usual analysis mode, the user clicks on "Complete FMEA" and an FMEA including every failure mode of every component is performed. In an interactive analysis mode, however, the users can click on individual components and select subsets of the failure modes. The interactive analysis is intended for evaluating the effects of design changes when a complete FMEA is not needed.

Figure 5 depicts the internal processing that takes place. To understand what the correct operation of the system should be, MPC performs a qualitative simulation using the causal model with all components in the NORMAL mode.

As outlined previously, the results of the qualitative simulation are not numbers, but rather qualitative descriptions of the outputs at those points. After saving the NORMAL simulation results, MPC iterates through the desired failure modes of the relevant components performing a qualitative simulation with each injected failure.



```
MICROPROCESSOR-11     (STUCK R12 HIGH)     OUT-VOUT     extraneous (extra) output
MICROPROCESSOR-11     INPUT-BIT-ERROR      SOFTWARE     nonfunctional
                                                        Executes erroneous instructions,
                                           OUT-VOUT     Uses data with one or more wrong bits
                                                        possible bad results propagated to output,
MICROPROCESSOR-11     OUTPUT-BIT-ERROR     OUT-VOUT     unpredictable behavior
MICROPROCESSOR-11     TOTAL-FAILURE        OUT-VOUT     incorrect output
SELECTOR-5            (STUCK O4 LOW)       OUT-VOUT     nonfunctional
SELECTOR-5            (STUCK O4 HIGH)      SOFTWARE     incorrect output
                                           OUT-VOUT     Receives erroneous data
SELECTOR-5            (STUCK O3 LOW)       OUT-VOUT     possible bad results propagated to output
                                                        extraneous (extra) output
```

Figure 4

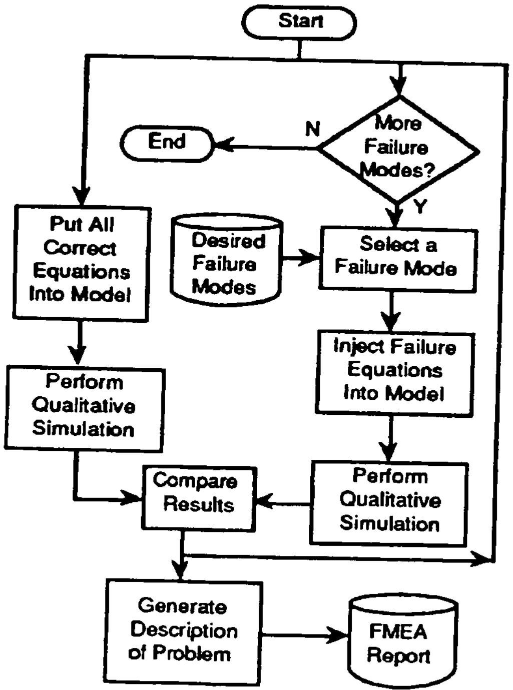

Start
More Failure Modes?
N
End
Y
Put All Correct Equations Into Model
Desired Failure Modes
Select a Failure Mode
Inject Failure Equations Into Model
Perform Qualitative Simulation
Perform Qualitative Simulation
Compare Results
Generate Description of Problem
FMEA Report

Figure 5

The qualitative simulation algorithms used by MPC are based on those described in Ref. 3. Rather than simulating with numbers, these algorithms simulate with qualitative values, such as {+,0,-}, in effect considering only the relative magnitudes of quantities. To determine the output behavior, these qualitative reasoning approaches construct transition diagrams showing how the qualitative values change over time. However, some serious shortcomings arise when using this simplistic approach. The worst problem is ambiguity (Ref. 7). For example, when subtracting a '+' from a '+', the result could be '+', '0', or '-'! Various improvements to the basic algorithms have been described in the qualitative reasoning literature (Ref. 8). For the automated FMEA, we selected an approach which replaces the static values such as {+,0,-} with symbolic values representing entire waveforms (Ref. 5). For example, the qualitative value F(decr,const) describes an oscillation of decreasing amplitude and constant frequency. In the FMEA analysis we found that reasoning about waveforms directly not only reduces ambiguity problems, but eliminates the need for translating time series of simulation results into waveform descriptions for the FMEA report.

Generating the FMEA Report

As depicted in Figure 5, the process of generating the FMEA report consists of comparing the faulted to the normal qualitative simulation results, and translating the differences into a short summary phrase. The style of reports generated by the prototype tool are qualitative, describing an effect on performance, rather than a numerical measure of degradation. For example, in the FMEA report shown in the interface screen of Figure 4, results such as "degraded output, phase inversion" and "output stuck at power supply voltage" are present. As our requirements analysis has shown, this is a predominant information type required by DOD and NASA.

The effect summaries are generated through a series of comparisons between the normal and failure simulation results. These comparisons include:

1. Is the value no longer affected by the same input parameters?

2. Is the qualitative waveform of the wrong type (or, if digital, is it the wrong function of its controlling parameters)?

3. If it is the correct waveform type, is there a phase inversion in the waveform?

4. If analog, is the value larger or smaller than it should be?

5. Is erratic behavior (noise) present on the output?

6. If no other problems noted, is the qualitative value different?

For each comparison, if the answer is yes, a corresponding English phrase is generated for the report. This phrase may be customized for particular customer requirements. If the answer to every question is no, the phrase "No degradation noted" is used. When the FMEA report is generated, it is first displayed on the screen, as shown in Figure 4. The analyst also has the option to save the report to a file using the "Save FMEA Report" option of the menu.

Applicability of the Automated FMEA Tool

The Automated FMEA Tool has been validated and tested with the interface in the domain of electronic circuits, as well as for a variety of microprocessor-based systems (Test results are reported in detail in the section entitled *"Testing of the Automated FMEA Tool"*). Additionally, in a more experimental mode, it has successfully analyzed small-scale-integration digital circuits and a hydraulic system. In theory, the FMEA tool should be equally applicable to any system that can be modeled for MPC. Characteristics of such systems include:

1. The system should be described as a set of components which have well-defined inputs and outputs.

2. The interactions between outputs of a component and inputs of other components should be known.

3. The cause/effect behavior of each component should be described only in terms of its inputs, outputs, and internal state; not in terms of internal state of other components.

4. The cause/effect behavior of each component should be described in terms f qualitative or quantitative equations or relations between discrete values.

Many classes of systems can be viewed in this way. Representative classes of systems which have been modeled in MPC include:

1. Electrical systems
2. Digital systems
3. Data Processing hardware
4. Data Processing software processes
5. Hydraulic systems
6. Robotic systems (manipulators, objects, tools)
7. Logistics systems (resources, transportation, users)

Classes of systems which will be more difficult, and which we have not yet attempted to model include:

1. Mechanical structures
2. Software at the code level (recursion, iteration)

Testing of the Automated FMEA Tool

It took several iterations between the software designer and reliability engineer to validate the accuracy of the Automated FMEA. An expert analyst was consulted to determine which techniques might best fit this application. After the tool was operational, several simple circuits were analyzed by the tool and by the expert to compare accuracy and the length of time to complete the analysis.

The first example was a circuit composed of one transistor, two resistors and one capacitor, implying a total of six failure modes (based on a limited failure mode library). It took the engineer 10 minutes to analyze the circuit by hand, and 52 seconds for the Automated FMEA Tool to build its model and complete the analysis with correct results. The second example was a slightly more complex two-stage amplifier composed of two transistors, five resistors and one capacitor, implying a total of 11 failure modes. It took the engineer 20 minutes to analyze the circuit by hand and 4 minutes and 50 seconds with the tool. The third example was a microcomputer system consisting of a microprocessor, memory chips, timing circuitry, and input/output devices. The schematic consisted of six VLSI chips, three SSI chips, and a quartz crystal clock module. A total of 25 failure modes were considered. It took the engineer 35 minutes to perform the FMEA by hand, whereas the tool required only 90 seconds. This shows an improvement in time of a factor of 2 to 50, despite the fact that the tool has not been optimized for speed.

These above times include only the time required for the FMEA tool to perform the analysis based on the schematic diagram already being loaded. We assume that in actual practice, the diagrams will have been previously generated in the design process and will be transferred electronically to the FMEA tool. For environments where this is not the case, the time required for entry of the schematic will also need to be considered. For the examples shown, this time was two minutes, four minutes, and 11 minutes, respectively. Even when these additional times are included, the automated FMEA exhibits a significant time savings.

For larger systems, we would expect an even greater improvement in efficiency. Additionally, there will be no variation in accuracy between engineers. The tool will always come up with the same results in the same amount of time, resulting in a better, more consistent analysis.

5. THE FUTURE

The tool has broad application potential. Our primary goal is to integrate the tool into the CAD/CAE environment. We have already successfully ported the MPC tool to a CAD platform. Our strategy is to migrate the code to C language, to improve the portability, and create appropriate CAD/CAE interface modules. Preliminary analysis indicates the tool will run on an IBM, or compatible, 386 type PC with sufficient memory.

Though the present tool is qualitative in nature, it can be integrated with quantitative analysis programs. Then, changes in part performance, like resistor drift, can be evaluated and performance thresholds established as failure criteria.

These techniques are applicable to any system whose operation can be described as a cause and effect relationship.

Once a model has been created for a system, and analyses performed, the data can be used by other MPC modules to create troubleshooting procedures, technical documentation, or command sequences to correct or work around problems.

The MPC module is a foundation upon which a total comprehensive RAMCAD system can be built.

ACKNOWLEDGMENT

The authors would like to acknowledge the work of Robin Kladke, a member of the automated FMEA team, who implemented many of the graphics features of the tool.

REFERENCES

1. Layne, J.D., "Improving Computational Reliability with Automatic Differentiation and Self-Validating Numerical Methods," **Proc. AIAA Computing in Aerospace 8 Conference, 1991.**

2. Gerald, C.F., **Applied Numerical Analysis**, Addison Wesley, 1978.

3. Bobrow, D.G. (editor), **Qualitative Reasoning about Physical Systems**, Elsevier Science Publishers, 1984.

4. Waterman, D.A., **A Guide to Expert Systems**, Addison-Wesley, 1986.

5. Schaefer, P., "Analytic Solution of Qualitative Differential Equations," **Proc. 9th National Conference on Artificial Intelligence (AAAI-91)**, pp. 830-836, 1991.

6. Iwasaki, Y., and Simon, H.A., "Causality in Device Behavior," **Journal of Artificial Intelligence**, Vol. 29, pp. 3-32, 1986.

7. Sacks, E.P., and Doyle, J., "A Prolegomena for any Future Qualitative Physics," **Technical Report CS-TR-314-91**, Princeton University Department of Computer Science, 1991.

8. Weld, D.S., and de Kleer, J. (editors), **Readings in Qualitative Reasoning about Physical Systems**, Morgan Kaufmann, 1990.

BIOGRAPHIES

Dan Bell
Martin Marietta Astronautics Group
P.O. Box 179
M.S. DC6073
Denver, CO 80201 USA

Mr. Bell received a Bachelor's Degree from UCLA in 1972, and received a Master of Science Degree in Systems Engineering Management from the University of Southern California in 1988. He worked at AiResearch Manufacturing Company in California until moving to Martin Marietta in August 1981. At AiResearch he worked in safety, reliability, and design engineering. He implemented stress/strength type analysis techniques and participated in the implementation of automated fault tree analysis techniques.

At Martin Marietta Mr. Bell is presently the Manager of the Reliability Organization for Strategic Systems in Denver. He has done extensive research into warranties, taught reliability in the Martin Marietta evening institute, lead efforts to implement SPC and related Total Quality Management initiatives and has been a major proponent of automation.

Lisa Cox
Martin Marietta Astronautics Group
P.O. Box 179
M.S. DC6073
Denver, CO 80201 USA

Ms. Cox received a Bachelor's Degree from Colorado School of Mines in 1987 and received a Master of Computer Information Systems with emphasis in Artificial Intelligence from the University of Denver in 1991. She has worked at Martin Marietta since 1988 in the Strategic Systems reliability group and has performed various reliability analyses and studies. She has developed several automated tools for reliability, maintainability and availability analyses as well as an allocation and a risk analysis tool.

Steve Jackson
Martin Marietta Astronautics Group
P.O. Box 179
M.S. DC6073
Denver, CO 80201 USA

Mr. Jackson received a Bachelor's Degree from the University of Colorado in 1989. He has worked at Martin Marietta since 1989 in the Strategic Systems reliability group. During that period he has performed FMEAs on mechanical systems as well as assisted in the development of reliability tools. For the last year he has developed flight dynamics performance simulation tools.

Phil Schaefer
Martin Marietta Astronautics Group
P.O. Box 179
M.S. 4372
Denver, CO 80201 USA

Mr. Schaefer received B.S. and M.S. in Electrical Engineering from Case Western Reserve University, 1986. He is currently working in the Artificial Intelligence group at Martin Marietta, and was technical lead in the development of the MPC tool. He has authored or co-authored 15 articles in the field of Artificial Intelligence.